UCRL-CONF-210371

Phase Sensitive Cueing for 3D Objects in Overhead Images

D. W. Paglieroni, W. G. Eppler, D. N. Poland

March 10, 2005

SPIE Defense & Security Symposium
Orlando, FL, United States
March 29, 2005 through April 1, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Phase Sensitive Cueing for 3D Objects in Overhead Images [1]

David W. Paglieroni [2], Walter G. Eppler [3] and Douglas N. Poland

ABSTRACT

A 3D solid model-aided object cueing method that matches phase angles of directional derivative vectors at image pixels to phase angles of vectors normal to projected model edges is described. It is intended for finding specific types of objects at arbitrary position and orientation in overhead images, independent of spatial resolution, obliqueness, acquisition conditions, and type of imaging sensor. It is shown that the phase similarity measure can be efficiently evaluated over all combinations of model position and orientation using the FFT. The highest degree of similarity over all model orientations is captured in a match surface of similarity values vs. model position. Unambiguous peaks in this surface are sorted in descending order of similarity value, and the small image thumbnails that contain them are presented to human analysts for inspection in sorted order.

Keywords: broad area search, model matching, pixel phase, match disambiguation, image thumbnail cueing

1. INTRODUCTION

The search for specific types of objects of unknown position and orientation in overhead images with potentially broad area coverage is of great importance to people that analyze imagery. The problem is difficult because images can be highly cluttered, and they can be acquired with different sensors, at different times of the day, in different seasons of the year, at various spatial resolutions, and with varying degrees of obliqueness. Computer systems that attempt to search images rapidly have been largely rejected by human analysts. Many attempts have been made to replace human analysts with computer systems that automatically detect specific types of objects in images. The problem is that given the maturity of existing algorithms, people lack confidence in the ability of computers to correctly interpret images and detect specific types of objects without supervision. Moreover, too little effort has been made to develop concepts for large scale image cueing systems that focus the attention of human analysts on locations that contain specific types of objects without eliminating humans from the loop.

This paper describes a two-stage computer-assisted approach for focusing human analyst attention on places in overhead images with potentially broad area coverage that contain specific types of objects. In the matching stage, a computer automatically matches models of 3D objects to overhead images one image block at a time using a novel fast algorithm based on pixel phase designed to handle variations in acquisition conditions and the type of imaging sensor. The strongest degree of match over all object orientations is computed at each position. Unambiguous local maxima in the degree of match as a function of pixel location are then found. In the cueing stage, a computer sorts image thumbnails in descending order of figure-of-merit and presents them to human analysts for visual inspection and interpretation. Thumbnail figure-of-merit is computed from degrees of match to a 3D object model associated with unambiguous local maxima within the thumbnail. This form of computer assistance is very useful when most of the relevant thumbnails are highly ranked, because the amount of inspection time needed is much less for relatively few highly ranked thumbnails than for entire images with large area coverage.

Section 2 describes how to project surfaces of solid models for 3D objects onto overhead images. It is argued that visible edges associated with projected surfaces are appropriate and often sufficient for model matching because unlike other pixels, model edges are often salient and visible in overhead images, independent of the imaging sensor and acquisition conditions.

Section 3 describes phase sensitive matching – a novel fast algorithm based on pixel phase for matching projected model edges to overhead images (both monochrome and multi-band). Pixel phase refers either to the direction of flow from dark to light in an image, or to the direction of the normal to a boundary curve in projected model edges. Phase sensitive matching rewards agreement in pixel phase between pairs of images. It is relatively insensitive to

[1] This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

[2] Lawrence Livermore National Laboratory paglieroni1@llnl.gov, poland1@llnl.gov
[3] Lockheed Martin IS&S walter.g.eppler@lmco.com

variations in image intensity because it deals primarily with directional (i.e., phase, as opposed to amplitude) information at the pixel level. It is somewhat related to a multi-source image auto-registration technique known as normalized cross-correlation of complex gradients (NCCCG), except NCCCG uses both intensity and phase information from the gradient at each pixel to match images acquired by different sensors ([1]-[2]). Phase sensitive matching is more closely related to the matching technique based on pixel phase in [3] used to automatically geo-register images by matching images to projections of 3D site model edges. This paper uses a similar approach to detect specific types of objects at arbitrary positions and orientations in overhead images.

Techniques for matching projected model edges to images can be categorized as those based on edges, those based primarily on pixel intensity (amplitude or luminance), and those based primarily on pixel phase (direction or orientation). Edge matching techniques require edges to first be extracted from images ([4]-[7]) so that projected model edges can be matched to them. Chamfer matchers compute average distances from image edge pixels to the nearest projected model edge pixels or vice-versa ([8]). Chamfer matchers are based on distance transforms of edge maps, i.e., arrays of distances from each pixel to the nearest edge pixel ([9]-[13]). When applied to object detection, edge matching techniques are known to be highly sensitive to edge clutter and the quality of detected edges ([14]).

Since one of the images is a map of projected model edges and the other is monochrome or multi-band, techniques that attempt to match pixel intensity require the monochrome or multi-band images to first be subjected to edge-enhancement pre-processing so that edges end up bright relative to the background. Normalized cross-correlation is a form of cross-correlation that is relatively insensitive to bias and gain in pixel amplitude ([15]-[17]). A related method based on correlation coefficients (which vary from −1 to 1) has properties similar to normalized cross-correlation ([18]). Phase correlators use the phase of the Discrete Fourier Transform (DFT) of the spatial correlation of two images ([19]-[20]). The translational offset between two images is chosen as the location of the maximum of the inverse DFT of the phase image. Matching techniques based primarily on pixel amplitude tend to work well when the two images being matched are comparable, as in the case of two images of an area acquired with the same sensor at nearly the same time. However, they often fail when the two images are highly disparate, as in the case of two images of an area acquired with different sensors, or an image and a line map of the same area ([1]).

Model matchers typically generate match surfaces that contain one match value per pixel. It is important to select certain points on the match surface as candidate object locations, and these points typically correspond to local maxima in the match surface. Match disambiguation techniques that distinguish between ambiguous and unambiguous local maxima are discussed in Section 4. Section 5 then describes how to use unambiguous local maxima to generate sorted lists of image thumbnails which reliably focus the attention of human analysts on places in images that contain specific types of objects.

2. OBJECT SPATIAL MODELS

Object models describe objects in terms of world coordinates *x*, *y* and *z*. For example, *x*, *y* and *z* could be the north, east and vertical position of vertices of straight line edges in a 3D object. Conceptually, vertex coordinates can be obtained from a blueprint of the object. In practice however, vertex positions are derived by photogrammetry from multiple images acquired from different viewpoints.

Object models are useful because they do not vary from image to image. They can be spatially modeled as collections of point, line, curve, shape and volumetric graphics primitives. Such primitives can be represented parametrically. For example, polygons can be represented with sequences of vertex coordinates, whereas cones and cylinders can each be represented with two points and one radius.

Forward projection is the process of mapping object space coordinates (i.e., world or geo-coordinates) [*x,y,z*] to image space (pixel) [*column,row*] coordinates [*c,r*]. Pixel coordinates [*c,r*] are traditionally expressed as *rational polynomials* [*C(x,y,z), R(x,y,z)*] of object space coordinates. These rational polynomials are of the form

$$(1) \qquad f(x,y,z) = \sum_{(i,j,k) \in N_f} a_f(i,j,k)\, x^i y^j z^k \Big/ \sum_{(i,j,k) \in D_f} b_f(i,j,k)\, x^i y^j z^k$$

where N_f and D_f are sets of distinct non-negative integer-valued exponents for the numerator and denominator polynomials. The headers of geo-registered images either explicitly contain the rational polynomial coefficients a_f and b_f, or parameters from which rational polynomial coefficients can be derived.

Within the local vicinity of a specific pixel in an overhead image, rational polynomial expressions for pixel [*column,row*] coordinates [*c,r*] can be approximated linearly using *affine models*, i.e., first-order linear combinations of object space coordinates [*x,y,z*]:

(2)
$$c = C(x,y,z) \approx a_{00}\,x + a_{01}\,y + a_{02}\,z + a_{03}$$
$$r = R(x,y,z) \approx a_{10}\,x + a_{11}\,y + a_{12}\,z + a_{13}$$

Affine models can be used to project spatially localized 3D objects onto specific locations within an overhead image. By using affine models as opposed to higher-order rational polynomials, the computational cost of forward projection can be significantly reduced. Two local linearization techniques for rational polynomials are Taylor Series Linearization and Least Squares Linearization.

Projections of object space model edges are formed by projecting each model surface onto the image (using established computer graphics techniques) and keeping only the surface edges. As each new surface is projected, edges from portions of previously projected surfaces that become occluded (hidden) are removed.

For a given image block, projected model edges are generated in three steps. In step 1, the model is rotated about the *z* axis through the model centroid in some angular increment. This step only needs to be performed once per object. In step 2, the rotated models are translationally offset such that the model centroid is shifted to the geo-coordinates of the pixel at the center of the image block. *Geopositioning (inverse projection)* is used to derive geo-coordinates from pixel coordinates. Geo-coordinates at an assumed height *z* can be quickly derived analytically from affine models or from rational polynomials using Newton iterations. If *z* is unknown but terrain elevation data is available, ray tracing can be used. In step 3, the translated and rotated models are projected onto the image block, retaining only the edges. Hidden surfaces and edges are removed using established computer graphics techniques.

3. PHASE SENSITIVE MATCHING

Phase sensitive matching is based primarily on the phase (i.e., the angle or direction) and mostly ignores the amplitude (luminance contrast) of directional derivatives in pixel intensity. Specifically, the degree of phase sensitive match between an image and projected model edges is based on the disparity between phases over all projected edge pixels and phases at corresponding image pixels. Phase sensitive matchers are designed for robust performance across images acquired by different sensors under diverse conditions. When the pattern to be matched is limited to edges, the matching results are based solely on pixels more likely to be salient regardless of image source, and this should promote consistency in matching results across images from different sources.

Phase sensitive matching works best on distinctive objects. An object can be distinctive due to its size, shape, or surface details. For example, a small rectangular object will not be distinctive in an image that contains large numbers of small regions that are nearly rectangular, but a large rectangular object might be very distinctive. However, a small rectangular object may be distinctive if it has unusual surface markings. Also, an object with an unusual shape might be distinctive even it is relatively small.

3.1 Phase Estimation

For monochrome or multi-band images *u*, the directional derivative $\dot{u}\,(c,r)$ at pixel [*c,r*] can be estimated as a weighted sum of direction vectors from the center of pixel [*c,r*] to the centers of neighbor pixels [*c′,r′*]. $\dot{u}\,(c,r)$ can be expressed as a complex number with amplitude *A*(*c,r*) and *pixel phase* (i.e., angle or direction) $\theta\,(c,r)$.

Let $R(c,r \mid \rho)$ be the *neighborhood* of pixel [*c,r*], i.e., the set of pixels, exclusive of [*c,r*], that intersect the circle with *neighborhood radius* $\rho = 1, 2 \ldots$ centered on pixel [*c,r*] :

(3)
$$R(c,r \mid \rho) \overset{\Delta}{=} \{\, [c',r'] \neq [c,r] : (\,|c' - c| - \tfrac{1}{2})^2 + (\,|r' - r| - \tfrac{1}{2})^2 < \rho^2 \,\}$$

$\dot{u}\,(c,r)$ can be expressed in the general form

$$\dot{u}(c,r) \overset{\Delta}{=} A(c,r)\, e^{j\theta(c,r)} = \sum_{(c',r')\in R(c,r\,|\,\rho)} [\, u(c',r') - u(c,r)\,] \cdot \frac{(c'-c) + j(r'-r)}{[(c'-c)^2 + (r'-r)^2]^{k/2}} \tag{4}$$

The directional derivative estimate $\dot{u}(c,r)$ has two parameters: the neighborhood radius ρ and the *distance power* $k \geq 0$ (i.e., the power on Euclidean distance from the center of pixel $[c,r]$ to the center of pixel $[c',r']$). This estimate has an equivalent representation that uses pairs of convolution kernels. One kernel is used to generate the real part of $\dot{u}(c,r)$, and the other generates the imaginary part. For example, when $\rho = 1$, the phase estimates are based on pixels within 3x3 neighborhoods, and the convolution kernels for the real and imaginary parts of $\dot{u}(c,r)$ are given by

$$\begin{bmatrix} -\alpha & 0 & \alpha \\ -1 & 0 & 1 \\ -\alpha & 0 & \alpha \end{bmatrix}, \quad \begin{bmatrix} -\alpha & -1 & -\alpha \\ 0 & 0 & 0 \\ \alpha & 1 & \alpha \end{bmatrix}, \quad \alpha = 2^{-k/2} \Leftrightarrow k = -2\log_2 \alpha \quad (k \geq 0,\ \alpha \leq 1) \tag{5}$$

For $\rho = 1$, it is clear that $k = 0$ corresponds to Prewitt ([4]) and $k = 2$ corresponds to scaled Sobel phase estimation ([5]). $k = 1$ represents a compromise between Prewitt and Sobel phase estimation. It can be readily shown that for $\rho = 1$, $k = 2$ is the only non-negative integer for which the phase estimate for ideal binary edges that pass through the centers of 3x3 neighborhoods is exact at angles of 0, $tan^{-1} 1/3 \approx 18.4°$ and $45°$, and the phase estimates are close to being exact between these angles. Similar results apply to the remaining adjacent $45°$ intervals because of symmetry in the convolution kernels. For $\rho = 1$, the phase estimates for $k = 2$ are thus exact at roughly every $20°$, and the estimates are close in between. This suggests that for $\rho = 1$, $k = 2$ (Sobel) is a good choice for phase estimation (i.e., it is more accurate in a meaningful sense than phase estimates based on other non-negative integer values of k).

Noise in the phase estimate can be expected to decrease as the neighborhood radius ρ increases, but the computational complexity of phase estimation increases. A small value of ρ should be used if the patterns to be matched contain finely spaced detail (e.g., we use $\rho = 1$ when matching object projections), whereas larger values should be used if the patterns contain mostly coarse level detail. For any non-negative integer ρ, the influence that individual neighbor pixels $[c',r']$ have on phase estimates decreases with proximity to $[c,r]$, and for proximities greater than 1 pixel, the degree of influence decreases as k increases. In the limit as $k \rightarrow \infty$, the only pixels that affect the phase estimate at $[c,r]$ are its four nearest neighbors. If $k = 0$ and $u(c',r') - u(c,r)$ is fixed, the direction vector $(c'-c) + j(r'-r)$ is uniformly weighted in equation (4), so neighbors $[c',r']$ farther from $[c,r]$ have a greater impact on $\dot{u}(c,r)$ when they should have less. If $k = 1$, the direction vector is a unit vector, so neighbors have the same impact on $\dot{u}(c,r)$ regardless of their distance from $[c,r]$. If $k = 2$, the impact of individual neighbors on $\dot{u}(c,r)$ varies inversely with distance from $[c,r]$, but collectively, the set of all pixels at fixed distance from $[c,r]$ has the same impact on $\dot{u}(c,r)$, independent of distance. However, if $k = 3$, the set of all pixels at fixed distance from $[c,r]$ collectively have an impact on $\dot{u}(c,r)$ that varies inversely with distance. $k = 3$ is thus a desirable choice for $\rho > 1$. A phase example is shown in Fig.1 for a football field image (courtesy of TerraServerUSA).

One way to generalize equation (4) to images with $B > 1$ bands is to write equation (4) for each band, sum the B equations into a single composite equation, and divide both sides of the composite equation by B. This is tantamount to interpreting $u(c,r)$ in equation (4) as the mean of values of pixel $[c,r]$ across all B bands. For multi-band images u, $u(c,r)$ in equation (4) can thus be interpreted as the value of pixel $[c,r]$ in a band-averaged version of u.

Phase is estimated differently for pixels on edge curves than for pixels in monochrome or multi-band images. Rather than using the angle of a directional derivative in pixel intensity, the angle of the normal to the edge or curve boundary is estimated at each edge pixel. For edge patterns derived directly from images (e.g., image space edge models traced over a displayed image with a mouse in the absence of 3D object space models), phase can be estimated as a weighted sum of angles of rays from the center of the edge pixel of interest to neighboring edges on the same curve (where the weights are non-negative and sum to one). This type of phase estimation is facilitated by using fully thinned edges. However, if 3D object models are available, it is better to estimate phase analytically as the model surfaces and edges are being projected, rather than discretely from projected edge pixels. For example, each pixel on a projected line segment has the same phase, namely the angle of the normal to the projected line segment, which can be computed analytically from the projected endpoints.

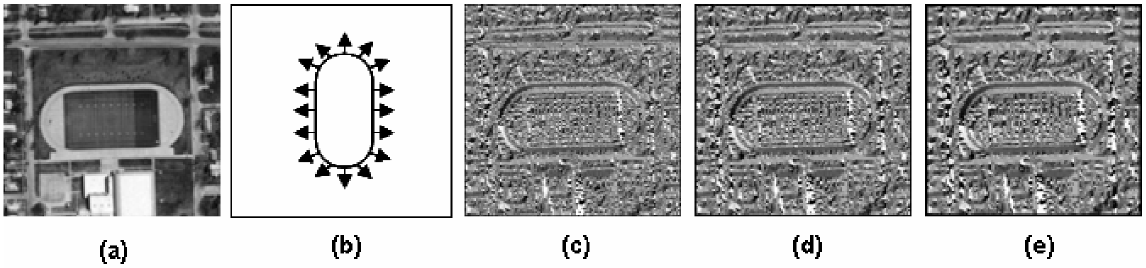


Fig.1 (a) Image of football field. (b) Phases for projected model edges of football field.
(c)-(e) Phase estimates for pixels in image of football field for $k = 3$ and $\rho = 1, 2$ and 3.

3.2 FFT-Based Matching Algorithm

Consider an image $\boldsymbol{\theta} \overset{\Delta}{=} \{\theta(c,r)\ \ c = 0\ldots w{-}1;\ \ r = 0\ldots h{-}1\}$ of phases for pixels in a monochrome or multi-band image block with w columns and h rows. Let $\boldsymbol{P}$ be a list of $N_P > 0$ projected model edge pixels for some object projected onto the image block at some orientation. Consider a second image $\boldsymbol{\beta} \overset{\Delta}{=} \{\beta(c,r)\ \ c,r = 0\ldots 2R_{max}\}$ of boundary phases for the bitmap of projected model edges associated with $\boldsymbol{P}$. Assume that boundary phase can be estimated unambiguously for each of these edge pixels. R_{max} is the model projection radius, i.e., the radius of the projection bounding circle defined as the pixel distance from the projected edge centroid to the most remote projected edge pixel (rounded up to the nearest integer).

In order to minimize the influence that acquisition conditions and the type of imaging sensor have on matching results, assign a fixed amplitude of $a(c,r) = 1$ to all pixels $[c,r]$ in the h x w image block, and set $a(c,r) = 0$ outside the block. Also, set $a(c,r) = 0$ at pixels $[c,r]$ for which the directional derivative amplitude ($A(c,r)$ in equation (4)) is less than some threshold A_{min}. This allows the matcher to only take into account image block pixels at which the contrast variation is perceptible. For example, if $\rho = 1$ and $k = 2$ in equation (4), an ideal vertical step edge with a barely perceptible gray level contrast of 6 corresponds to a critical amplitude of $A_{min} = 12$.

In the absence of additional information, the same weight $b(c,r)$ should be assigned to all projected model edge pixels $[c,r] \in \boldsymbol{P}$. If uniform non-negative weights that sum to unity are required, then $b(c,r) = 1 / N_P$ for all $[c,r] \in \boldsymbol{P}$. However, in some phase sensitive matching problems, it may be appropriate to assign different non-negative weights that sum to unity to different pixels (a topic for future research).

Let us require the similarity $S(\Delta_c,\Delta_r)$ between phase image $\boldsymbol{\beta}$ at an offset of $[\Delta_c,\Delta_r]$ and phase image $\boldsymbol{\theta}$ to vary from 0 (for poor similarity) to 1 (for perfect similarity). Such a measure of similarity can be derived by noticing that the square of the cosine of the difference between two angles β and θ varies from 0 (for angles of vectors pointing in orthogonal directions) to 1 (for angles of vectors pointing in the same or opposite directions). A similarity measure that satisfies the design requirements is

$$(6)\qquad S(\Delta_c,\Delta_r) = \frac{1}{2} + \sum_{(c,r)\in \boldsymbol{P}} a(c+\Delta_c,r+\Delta_r)\, b(c,r) \left[cos^2\, [\theta(c+\Delta_c,r+\Delta_r) - \beta(c,r)] - \frac{1}{2} \right]$$

$$= \frac{1}{2} + \frac{1}{2} \sum_{(c,r)\in \boldsymbol{P}} a(c+\Delta_c,r+\Delta_r)\, b(c,r)\, cos\, 2[\theta(c+\Delta_c,r+\Delta_r) - \beta(c,r)]$$

The design requirements are still met even if both terms of "*1/2* " from the first expression in equation (6) are removed. However, as shown below, the motivation for including both terms of "*1/2* " is to enable the similarity measure to be evaluated with one (as opposed to two) 2D convolutions. $S = 1$ corresponds to a perfect phase match between the image and projected model edges, whereas the expected degree of match between an image with uniformly distributed random phases and projected model edges is $S = 1/2$. $S = 0$ occurs when image phase is orthogonal to projected model edge phase at every edge pixel (this would be a highly unusual situation).

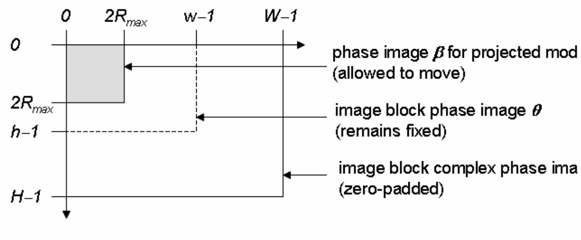
In equation (6), it is understood that the dimensions of $\boldsymbol{\theta}$ (namely w and h) must be at least as large as the width of $\boldsymbol{\beta}$ (namely $2R_{max}+1$). The range of admissible offsets is thus $\Delta_c = 0 \ldots w - 2R_{max} - 1$ and $\Delta_r = 0 \ldots h - 2R_{max} - 1$ (w, $h > 2R_{max}$). Note that an offset of $\boldsymbol{\beta}$ by $[\Delta_c, \Delta_r]$ corresponds to a projection centroid location of $[\Delta_c + R_{max}, \Delta_r + R_{max}]$ in the image block.

In the spatial domain, the computational complexity associated with evaluating $S(\Delta_c, \Delta_r)$ over all projection positions and orientations within an image block is directly proportional to the number of edge pixels (N_P), and can be large even for modest N_P. Fortunately, $S(\Delta_c, \Delta_r)$ can be efficiently evaluated over all projection positions and orientations using the Fast Fourier Transform (FFT). Since most FFT implementations are radix 2, let us zero-pad the phase images $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ to a width of W columns and a height of H rows of pixels, where W and H are the first powers of 2 $\geq w$ and h. To maximize computational efficiency, the image block dimensions w and h can be deliberately specified by the user as powers of 2 (in which case $W = w$ and $H = h$). Now consider two complex zero-padded images of size H x W:

(7a)
$$\boldsymbol{\Theta} \overset{\Delta}{=} \{\Theta(c,r) = a(c,r)\, e^{j2\theta(c,r)} \quad c = 0 \ldots W{-}1;\ r = 0 \ldots H{-}1\}$$

(7b)
$$\boldsymbol{B} \overset{\Delta}{=} \{B(c,r) = b(c,r)\, e^{j2\beta(c,r)} \quad c = 0 \ldots W{-}1;\ r = 0 \ldots H{-}1\}$$

The block geometry is depicted graphically in Fig.2.



0
$2R_{max}$
w–1
W–1
0
$2R_{max}$
h–1
H–1
phase image β for projected mod
(allowed to move)
image block phase image θ
(remains fixed)
image block complex phase ima
(zero-padded)

Fig.2 Geometry for FFT-based phase sensitive matching.

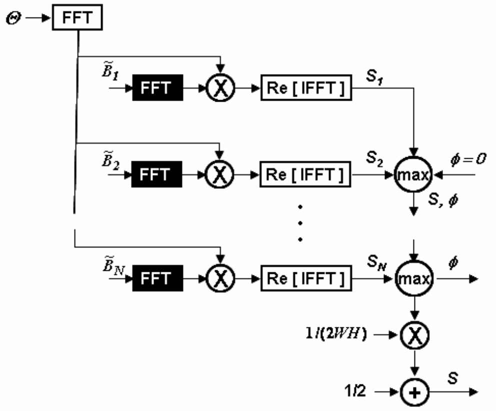


Fig.3 Block diagram for FFT-based phase sensitive matching. All DFT's and IDFT's are unscaled. The shaded boxes correspond to computations that may not need to be repeated from image block to image block.

From equations (6)-(7),

(8)
$$S(\Delta_c, \Delta_r) = \frac{1}{2} + \frac{1}{2} Re \left[\sum_{(c,r) \in \boldsymbol{P}} a(c+\Delta_c, r+\Delta_r)\, b(c,r)\, e^{j2[\theta(c+\Delta_c, r+\Delta_r) - \beta(c,r)]} \right]$$

$$= \frac{1}{2} + \frac{1}{2} Re \left[\sum_{c=0}^{W-1} \sum_{r=0}^{H-1} b(c,r)\, e^{-j2\beta(c,r)}\, a(c+\Delta_c, r+\Delta_r)\, e^{j2\theta(c+\Delta_c, r+\Delta_r)} \right]$$

$$= \frac{1}{2} + \frac{1}{2} Re \left[B^*(\Delta_c, \Delta_r) \circledast \Theta(\Delta_c, \Delta_r) \right]$$

$$= \frac{1}{2} + \frac{1}{2} Re \left[B^*(-\Delta_c,-\Delta_r) \ * \ \Theta(\Delta_c,\Delta_r) \right]$$

$$= \frac{1}{2} + \frac{1}{2WH} Re \left[\text{IDFT [DFT [}B^*(-\Delta_c,-\Delta_r)\text{]} \cdot \text{DFT [}\Theta(\Delta_c,\Delta_r)\text{]]} \right]$$

where "$*$" and "$\circledast$" are the 2D spatial convolution and correlation operators, B^* is the complex conjugate of B, B and Θ are assumed periodic with period W in the c dimension and H in the r dimension, and the circular convolution theorem of the DFT has been used. In equation (8), "DFT" and "IDFT" denote un-scaled forward and inverse two-sided (2D) DFT's, which can be efficiently evaluated using an FFT algorithm. The values of $S(\Delta_c,\Delta_r)$ computed with equation (8) are valid only for integers $0 \leq \Delta_c < w - 2R_{max}$ and $0 \leq \Delta_r < h - 2R_{max}$ (these are the admissible offsets in the discussion following equation (6)).

Equation (8) mathematically characterizes FFT-based phase matching between an image block and an object at one projection orientation. In general, matching must be performed for each of N projection orientations to produce matrices of match similarities $\boldsymbol{S_n}$ $n = 1 \ldots N$. For each admissible offset $[\Delta_c,\Delta_r]$,

(9) $$S(\Delta_c,\Delta_r) = \max_{n = 1 \ldots N} S_n(\Delta_c,\Delta_r)$$

and the angle $\phi(\Delta_c,\Delta_r)$ of best match is saved. Fig.3 gives a block diagram for FFT-based phase matching across all orientations, where $\tilde{\boldsymbol{B}}_{\boldsymbol{n}}$ be the complex conjugate of $\boldsymbol{B}$ in equation (7b) for the projection at orientation n folded about both the rows and columns axis

3.3 The Number of Object Orientations and Image Spatial Resolution

At an image spatial resolution of 1X, the number of required object orientations (N) increases with the size of the projection. If R_{max} is the radius of the bounding circle for projected model edges over all object orientations, N never needs to exceed the circumference $2\pi R_{max}$ of the bounding circle (in pixels). Let $\bar{R}$ be the mean distance from the projection centroid to the projected model edges over all object orientations. If $\bar{R}$ is significantly less than R_{max}, it is more appropriate to specify N as directly proportional to $\bar{R}$. By choosing $N = int\,(2\pi\bar{R}/k)$, projected edge pixels move roughly k pixels on average between successive rotations. If the phase estimates are derived from pixels within 3x3 neighborhoods, it makes sense to choose $k = 2$, in which case,

(10) $$N = int\,(\pi\bar{R}) \quad \Rightarrow \quad \Delta\theta \approx 2\,/\,\bar{R}$$

The computational cost (and in some implementations, the memory requirement) associated with phase sensitive matching increases linearly with the number of object orientations. Independent of image spatial resolution and object size, the computational cost of phase sensitive matching per image block can be driven to a fixed value by matching objects at the image spatial resolution for which the projections have a targeted mean radius of $\bar{R}_T$. It is assumed that although matching performance tends to improve with image spatial resolution, there is a critical resolution beyond which there will be no further improvement, namely the resolution at which the object projections are of a certain size. The critical spatial resolution is thus expected to be lower for larger objects. For $\bar{R} \geq \bar{R}_T$, the image block should be processed at a spatial resolution of αX, where

(11) $$\alpha = \bar{R}_T / \bar{R}$$

(use $\alpha = 1$ for $\bar{R} < \bar{R}_T$). We typically match objects at spatial resolutions for which $\bar{R}_T \approx 24$ pixels, in which case $N = 75$ and $\Delta\theta \approx 4.8°$. Reduced resolution images can be generated using, for example, bicubic interpolation. Objects to be matched are then projected onto the reduced resolution image.

Square image blocks matched at spatial resolutions of αX have a fixed width of w_α, which should b be chosen as a power of 2 when a radix 2 FFT is used (we use $w_\alpha = 512$). At 1X, these image blocks have width $w = int\,(w_\alpha / \alpha)$. The computational complexity of phase sensitive matching is

(12) $$\mathrm{O}[8N\, w_\alpha^2\, log_2 w_\alpha]\ \mathrm{OPS\ /\ block}\ =\ \mathrm{O}[8N\alpha^2 log_2 w_\alpha]\ \ \mathrm{OPS\ /\ pixel}$$

If the DFT's shown shaded in Fig.3 are disk-cached from block to block, then the memory requirement for FFT-based phase sensitive matching is $8(N{+}2)w_\alpha^2$ bytes (this assumes that the FFT computations are carried out in 8 byte double precision, but the results are stored in 4 byte single precision). For $N = 75$ and $w_\alpha = 512$, this amounts to ~160 MBytes.

3.4 Match Surface Examples

Fig.4 shows a 512x512 section of an image of a prison in Calipatria, CA (courtesy of TerraServerUSA). Fig.5 shows object model edge projections for "notched", "long" and "grated" buildings to be matched to the prison image. The projections were formed in image space by drawing edges with a mouse over one instance of each object in the image. This is appropriate here because our purpose is simply to show examples of match surfaces. Moreover, although the example image is ortho-rectified, it does not come with coefficients that relate pixel coordinates to geo-coordinates. Fig.6 shows surfaces $\boldsymbol{S}$ of match similarities for the models of the three types of buildings in Fig.5 against the prison image in Fig.4. The similarities range from 0 (dark) to 1 (light). Each model was matched at the image block spatial scale factor α specified in the previous section, and these scale factors are reflected in the sizes of the resulting match surfaces. For $\bar{R}_T = 20$, the values of α computed for the notched, long and grated buildings were 0.769, 0.513 and 1.0. The black regions surrounding the match surfaces occur because valid similarities only occur within a range of offsets $[\Delta_c, \Delta_r]$ that shrinks as the model projection radius increases (see the discussion following equation (6)).

4. MATCH DISAMBIGUATION

Match disambiguation is the process of extracting a complete set of unambiguous matches from a surface $S(c,r)$ of match similarities. A match at pixel $[c,r]$ is unambiguous if there is no pixel $[c',r']$ for which $S(c,r) \le S(c',r')$ and the matches at $[c,r]$ and $[c',r']$ are in conflict. Two criteria for conflicting matches are discussed in this section. In *conflict by proximity*, two matches conflict of their projected edge centroids lie within a prescribed distance (typically in pixels) of each other. Disambiguation by proximity is useful for finding objects that are expected to occur sparsely. In *conflict by overlap*, two matches conflict if their model projections overlap. To eliminate confusion between overlap and occlusion, the projections used to establish match ambiguity must be generated from models that have been flattened to a plane of constant z (height) value. Disambiguation by overlap is useful for finding objects that can be close together.

For each image, matching results can be stored in a file that contains a list of state vectors for disambiguated matches. These are feature vectors that contain the type of object, match pixel location, optimal match orientation, and match similarity value (from 0 to 1).

In *disambiguation by proximity*, a match $S(c,r)$ at pixel $[c,r]$ is unambiguous if and only if $S(c,r) > S(c',r')$ $\forall\ [c',r'] \in R(c,r \mid \Delta)$, where Δ is some user-specified Euclidean distance (pixels), and $R(c,r \mid \Delta)$ is the set of all pixels whose centers lie within Δ pixels of the center of pixel $[c,r]$. Algorithms for disambiguating lists and surfaces of matches by proximity are somewhat different. List-based disambiguation by proximity is computationally efficient only for short lists of matches, such as lists of sparsely separated matches from a match surface. Long lists of densely spaced matches are more efficiently disambiguated using algorithms that apply directly to match surfaces.

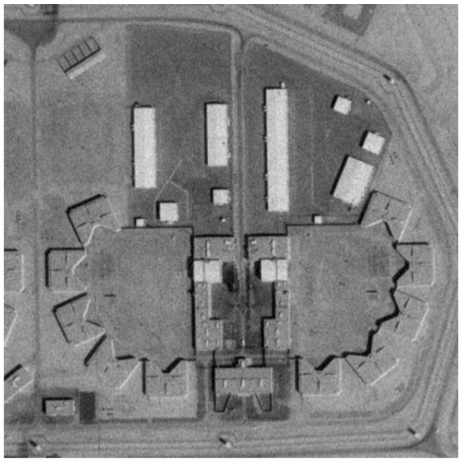


Fig.4 512x512 section of prison image for matching experiments.

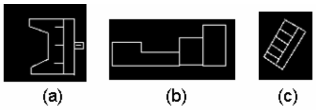


Fig.5 Image space model edge projections for three types of buildings in the prison image: (a) notched building (b) long building (c) grated building.

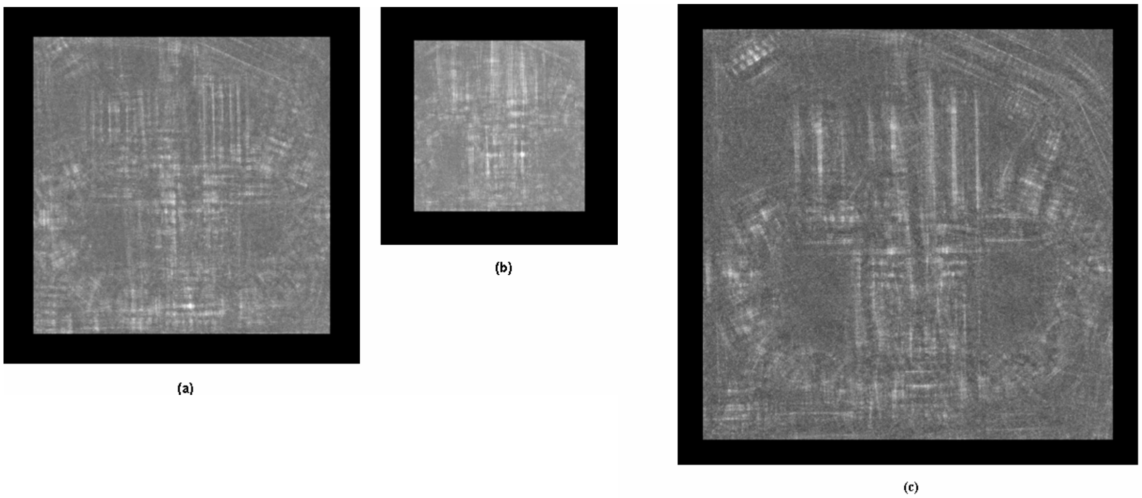


Fig.6 Surfaces of match similarities (from dark 0 to light 1) for the three models (a)-(c) in Fig.4 against the prison image in Fig.3.

In list-based disambiguation by proximity Δ, the image is divided into adjacent virtual tiles of width Δ, and a sub-list of matches is extracted for each tile. Each match M is compared (in terms of similarity value and proximity) only to the matches that lie in the 3x3 tile neighborhood of the tile that contains M.

In surface-based disambiguation by proximity Δ, a copy S' of the match surface S is made, and $\Delta_2 \leftarrow max(1, \Delta/2)$. For each pixel $[c,r]$ in S', if $S'(c,r) > 0$, then the maximum S_{max} of S is found for $(c',r') \in R(c,r \mid \Delta_2)$. If the maximum is not unique, it is set to some value > 1. Then, for $(c',r') \in R(c,r \mid \Delta_2)$, if $S(c',r') < S_{max}$, then $S'(c',r')$ is set to 0. The partially disambiguated matches occur at $(c,r) : S'(c,r) > 0$. The list of partially disambiguated matches is disambiguated by proximity Δ, as described earlier.

In *disambiguation by overlap*, a match $S(c,r)$ at pixel $[c,r]$ is unambiguous if and only if $S(c,r) > S(c',r')$ at all $[c',r']$ for which the projections of flattened models at $[c,r]$ and $[c',r']$ overlap. The process of determining if projections P_1 and P_2 (with n_1 and n_2 edge pixels respectively) overlap can be reduced from $O[n_1 n_2]$ complexity to $O[n_1+n_2]$ complexity by rasterizing one of the lists. Disambiguation by overlap can be computationally expensive if matches at each pixel on the match surface are compared to matches at every other pixel. The cost can be reduced by recognizing that projections of flattened models cannot overlap if their centroids are more than $2R_{max}$ pixels apart. The cost can be further reduced by first disambiguating the match surface by a proximity of $\Delta = max\ (\alpha\bar{R},\ \Delta_{min})$ for some sufficiently small Δ_{min} (say Δ_{min} = 3 pixels). Even for $\Delta = \Delta_{min}$, disambiguation by proximity will eliminate the majority of matches that need to be considered for disambiguation by overlap. Although some objects that lie close to each other might be missed, at least one of the objects in the area should be unambiguously matched, and only one such object needs to be matched in order for a human analyst cue to be successfully generated for the area.

Fig.7 shows examples of disambiguated matches of the long building in Fig.5(b) to the prison image in Fig.4 as colored dots. Examples are given for disambiguation by proximities of 128 and 32 pixels, and disambiguation by overlap.

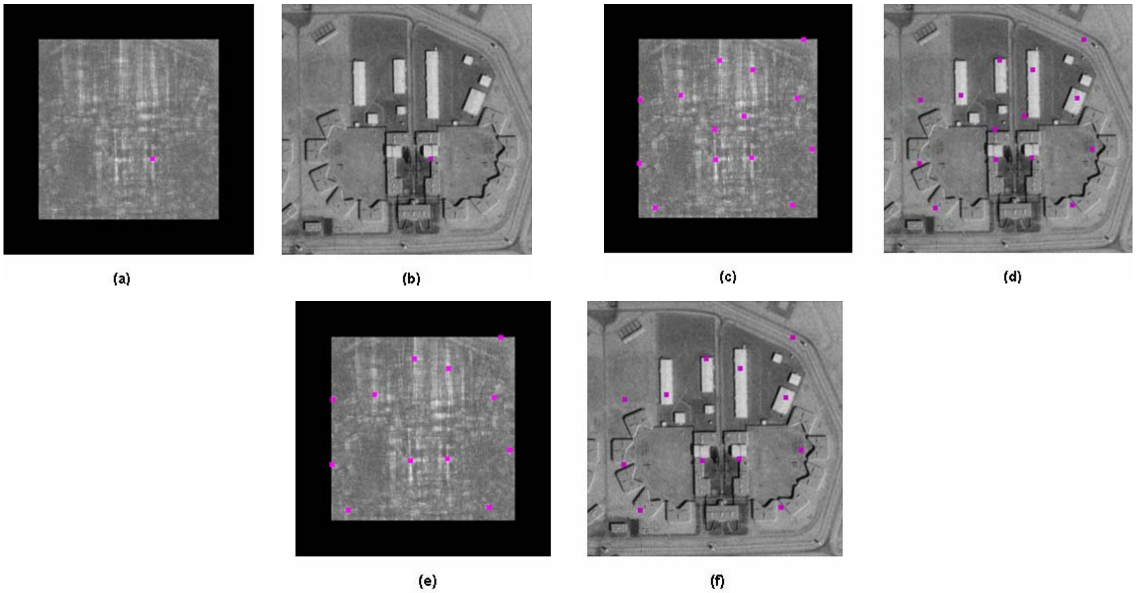


Fig.7 Matches (colored dots) of the long building from Fig.5(b) to the image in Fig.4 disambiguated by (a)-(b) a proximity of 128 (c)-(d) a proximity of 32 (e)-(f) overlap.

5. PHASE SENSITIVE CUEING WITH IMAGE THUMBNAILS

Three basic approaches to searching for prescribed objects in large volumes of overhead imagery are manual, automated and computer-assisted. In manual search, human analysts run application software for image display and manipulation on desktop computer workstations. Human analysts specify the order in which the images are manually searched. This approach will fail when human analysts are unable to meet search timelines.

Automated analysis is performed without human interaction. Computers are expected to not only match object models to images, but also decide which matches correspond to valid detections by interpreting the results. This approach is viable only if humans have confidence in the interpretation algorithms.

Computer-assisted analysis is a hybrid approach in which computers are expected to match object models to images (as in automated analysis), and human analysts are expected to interpret the results (as in manual analysis). This approach can work in situations where both fully manual and fully automated analysis fail, as long as computers are able to use the model matches to provide cues that focus human analyst attention on appropriate locations.

Fig.8 shows the strongest disambiguated phase sensitive matches to the three models in Fig.5 for a 1024x1024 section of an image of a prison in Calipatria, CA (courtesy of TerraServerUSA). Note that all buildings of each type were detected (the two apparently undetected long buildings are actually mirror images of the model and should not be detected). The results in Fig.8 were obtained by applying a decision threshold (in this case, a *similarity threshold* S_0) to the disambiguated matches. It is clear from Fig.8 that there exists a set of decision thresholds for which the detection probability is one and the false alarm rate is zero. The similarity thresholds that produced the results in Fig.8 were S_0 = 0.8, 0.82 and 0.72 for the notched, long and grated buildings respectively. The problem with attempting to automate the phase sensitive detection process is that one cannot know a priori what similarity thresholds to use, since appropriate values can vary considerably depending on the type of object, type of imaging sensor, and image acquisition conditions. A human-interactive mechanism is needed.

A more robust way to address the search problem is to treat it as a computer-assisted human-interactive cueing problem rather than as an automated object detection problem. One way to perform cueing is to divide the images to be searched into contiguous image thumbnail tiles of fixed size, compute a figure-of-merit (cueing metric) for each thumbnail based on the disambiguated matches that it contains, and sort the thumbnails in descending order of figure-of-merit. A human analyst can then visually inspect and interpret thumbnails near the beginning of the list. In this context, image thumbnails near the top of the list are cues that serve to focus the attention of human analysts on specific locations in the set of images to be searched. These thumbnails are presumably the most likely to contain specific objects of interest.

Even if the images to be searched are large and there are many of them, it may take only a few seconds per image to sort the thumbnails because the thumbnail figure-of-merit calculations are based on disambiguated matches that were previously extracted. It is true that large images may have to be matched one block at a time, and that parallel processing may be needed to meet time constraints for object matching in large sets of large images. However, thumbnail cueing often requires only one processor, and the time needed for a human analyst to inspect thumbnails near the top of the sorted list should be very much less than the amount of time needed to inspect the images in their entirety.

Fig.9 shows a display generated by an image thumbnail cueing application for the 1024x1024 prison image in Fig.8 and the grated building model in Fig.5(c). The thumbnails on the right are sorted in row-wise descending order of figure-of-merit. Notice that the first two thumbnails contain the two grated buildings that actually occur in the image. When a human analyst clicks on a thumbnail, the surrounding context for that thumbnail is displayed in the larger window on the left. The context for the first thumbnail is displayed in Fig.9. The number of thumbnails in the list is limited by a user-specified lower bound on figure-of-merit for image thumbnail cues.

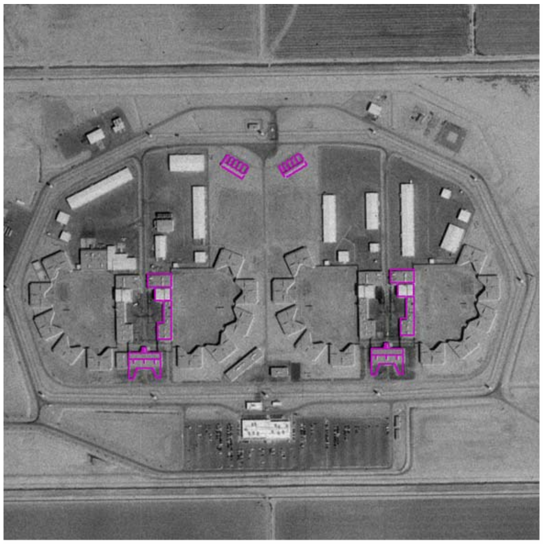
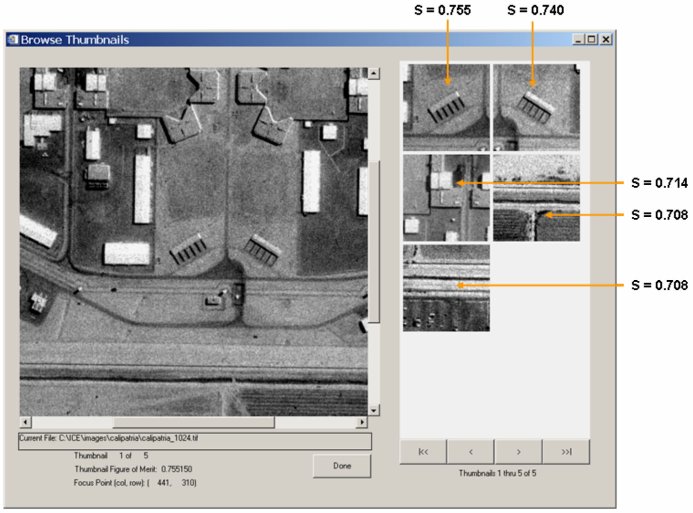


Fig.8 Renderings of the strongest disambiguated phase sensitive matches to the three models in Fig.5 for a 1024x1024 section of a prison image.



Current File: C:\ICE\images\calipatria\calipatria_1024.tif

Thumbnail 1 of 5

Thumbnail Figure of Merit: 0.755150

Focus Point (col, row): (441, 310)

Thumbnails 1 thru 5 of 5

Fig.9 Thumbnail cue results for the grated building in Fig.5(c) against the 1024x1024 prison image in Fig.8.

6. SUMMARY, CONCLUSIONS AND TOPICS FOR FUTURE RESEARCH

A similarity measure that matches phase angles of directional derivative vectors at image pixels to phase angles of vectors normal to edges of object models projected onto images at various positions and orientations was introduced. It is designed to be relatively insensitive to brightness and contrast, spatial resolution, imaging geometry, image acquisition conditions, and the type of imaging sensor used. A computationally efficient FFT-based method for evaluating the degree of model match over all object positions and orientations within an image block was developed. Efficient techniques for finding unambiguous peaks in the match surface formed from the similarity for the orientation of best match at each pixel location were then described. Finally, a computer-assisted cueing system for image search that creates lists of image thumbnails sorted in order of relevance based on the disambiguated matches they contain was proposed and demonstrated.

The theme of our proposed topics for future research is techniques that improve cueing efficiency. One major area for future investigation is techniques for assigning weights to projected edge pixels based perhaps on interactive training or relevance feedback mechanisms. Another proposed thrust involves techniques for merging lists of sorted image thumbnails from different images by combining various types of information from images and other sources. Although the amount of time it takes to search a cued image using our technique currently does not depend on image size, it does increase linearly with the number of images. This behavior can be curtailed by merging lists of image thumbnails across images.

REFERENCES

1. W. G. Eppler, D. W. Paglieroni, S. M. Petersen, M. J. Louie, "Fast Normalized Cross-Correlation of Complex Gradients for Autoregistration of Multi-Source Imagery", Proc. ASPRS DC 2000 Conf., May 22-27, 2000.
2. W. G. Eppler, D. W. Paglieroni, S. M. Petersen, M. J. Louie, "Normalized Crosscorrelation of Complex Gradients (NCCCG) for Image Autoregistration", U.S. Patent 6,519,372, issued February 11, 2003, assignee: Lockheed Martin.
3. W. G. Eppler, B. Trusso, “System for Registering Site Models to Gray-Scale Images”, unpublished white paper produced under contract NIMA NMA 201-01-C-0013, AFE/CD Topic 2, Autonomous Image to Model Registration, December 2001
4. J. M. S. Prewitt, “Object Enhancement and Extraction”, in Picture Processing and Pyschopictorics, B. S. Lipkin and A. Rosenfeld, eds., Academic Press, New York, 1970.
5. L. S. Davis, “A Survey of Edge Detection Techniques”, CGIP, Vol.4, 1975, pp.248-270.
6. D. Marr, E. Hildreth, “Theory of Edge Detection”, Proc. Roy. Soc. London, B207, 1980, pp.187-217.
7. J. Canny, “Computational Approach to Edge Detection”, IEEE Trans. PAMI, Vol.8, No.6, November 1986, pp.679-698.
8. H. G. Barrow, J. M. Tenenbaum, R. C. Bolles , H. C. Wolf, “Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching”, Proc. 5th Int. Joint Conf. Artif. Intell., Cambridge, MA, 1977, pp.659-663.
9. A. Rosenfeld, J. L. Pfaltz, “Sequential Operations in Digital Picture Processing”, J. Assoc. Comput. Mach., Vol.13, 1966, pp.471-494.
10. P. E. Danielsson, “Euclidean Distance Mapping”, CGIP, Vol.14, 1980, pp.227-248.
11. H. Yamada, “Complete Euclidean Distance Transform by Parallel Operation”, Proc. 7th Int. Conf. Pattern Recog., Montreal, Canada, 1984, pp.69-71.
12. G. Borgefors, “Hierarchical Chamfer Matching: a Parametric Edge Matching Algorithm”, IEEE Trans. PAMI, Vol.10, No.6, 1988, pp.849-865.
13. D. W. Paglieroni, “A Unified Distance Transform Algorithm and Architecture”, Machine Vision and Applications, vol.5, 1992, pp.47-55.
14. D. W. Paglieroni, G. E. Ford, E. M. Tsujimoto, “The Position-Orientation Masking Approach to Parametric Search for Template Matching”, IEEE Trans. PAMI, Vol.16, No.7, July 1994, pp.740-747.
15. B. V. K. Kumar, F. Dickey, J. DeLaurentis, “Correlation Filters Minimizing Peak Location Errors”, J. Opt. Soc. A., Vol.0, No.5, May 1992.
16. L. G. Brown, “A Survey of Image Registration Techniques”, ACM Comput. Surveys, Vol.24, 1992, pp.325-376.
17. L. Fonseca, B. S. Manjunath, ‘Registration Techniques for Multisensor Remotely-Sensed Imagery”, Photogram. Eng. Remote Sensing, Vol.62, 1996, pp.1049-1056.
18. M. Svedlow, C. D. McGillem and P. E. Anuta, “Experimental Examination of Similarity Measures and Pre-Processing Methods used for Image Registration”, Symposium on Machine Processing of Remotely Sensed Data, Westville, Indiana, June 1976, pp.4A-9.
19. C. D. Kuglin, D. C. Hines, “The Phase Correlation Image Alignment Method”, Proc. IEEE 1975 Int. Conf. Cybernetics and Society, IEEE, New York, September 1975, pp.163-165.
20. F. DeCastro and C. Morandi, “Registration of Translated and Rotated Images Using Finite Fourier Transforms”, IEEE Trans. PAMI, Vol.9, No. 5, September 1987, pp.700-703.